

# C언어 기초 프로그래밍 주말

김종국

QnA 메일 : [edujongkook@gmail.com](mailto:edujongkook@gmail.com)

Pdf 파일 : [github.com/edujongkook](https://github.com/edujongkook/pdf_sbs_c_weekend)  
[/pdf\\_sbs\\_c\\_weekend](https://github.com/edujongkook/pdf_sbs_c_weekend)

# 강의 일정

주	내용
1-2주	<b>CHAPTER 1 / 오리엔테이션 C 언어 소개:</b> c언어의 기본 구성요소들 / 데이터 형, 연산자, 수식
3-4주	<b>CHAPTER 2 - 프로그램 제어흐름:</b> 조건문 / 반복문을 이용한 함수 프로그래밍
5-6주	<b>CHAPTER 3,4 - 포인터와 배열, 문자열:</b> 포인터와 배열 / 문자열
7-8주	<b>CHAPTER 5 - 메모리 관리와 구조체:</b> 포인터 심화, 메모리 동적할당, 구조체

# 1주차

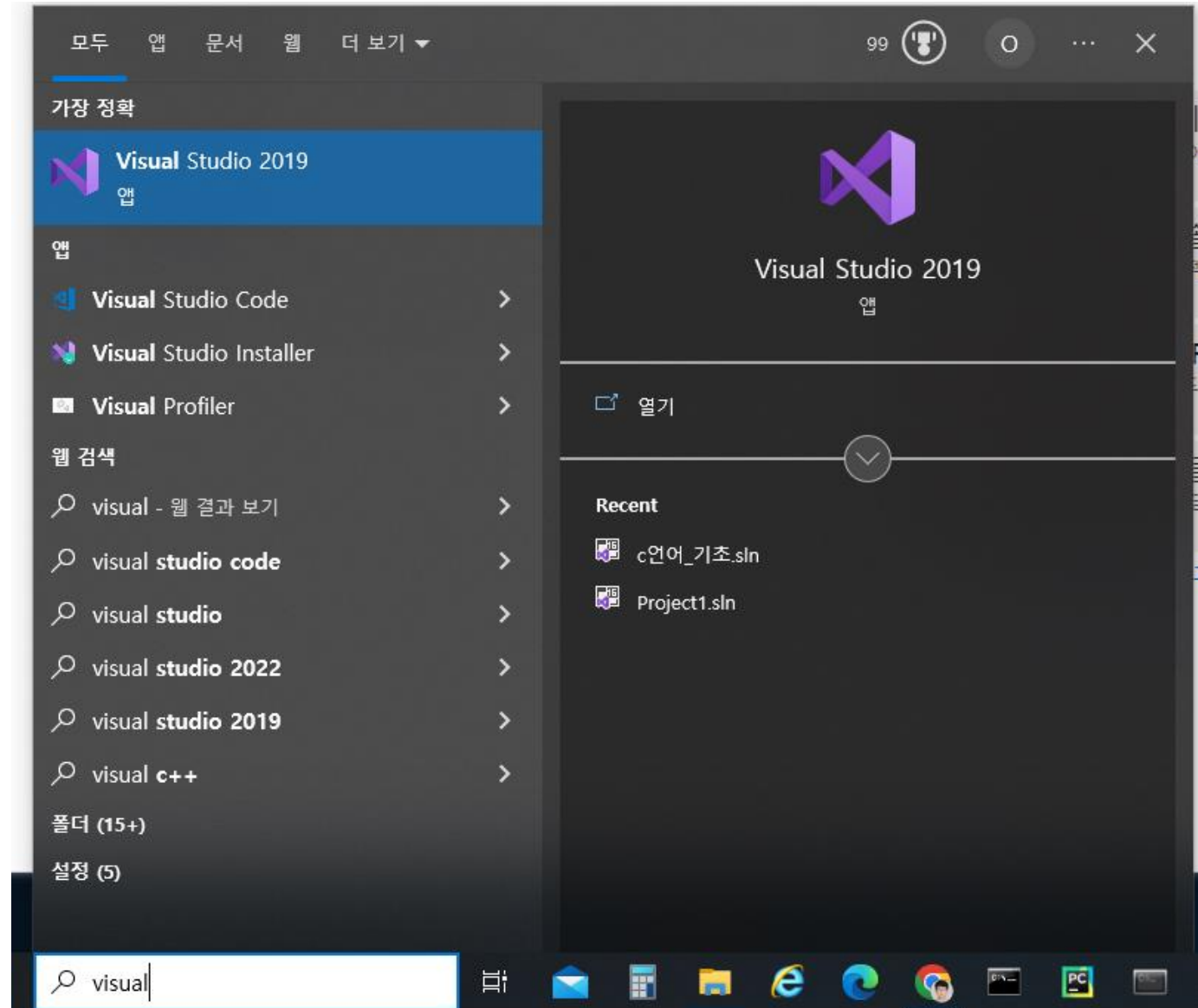
오리엔테이션

C언어  
기본 구성요소

IDE (Integrated Development Environment) 통합 개발 환경

# Visual Studio 2019

C 언어 개발 도구  
다운로드



# Visual Studio 2019

## 최근 파일 열기(R)

최근 항목 검색(Alt+S)(S)



### ▲ 어제



c언어\_기초.sln

2023-07-13(목) 오전 11:35

C:\Users\Worigener\Desktop\sbs\_c\c언어\_기초

## 시작



### 리포지토리 복제(C)

GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기



### 프로젝트 또는 솔루션 열기(P)

로컬 Visual Studio 프로젝트 또는.sln 파일 열기



### 로컬 폴더 열기(F)

폴더 내에서 탐색 및 코드 편집



### 새 프로젝트 만들기(N)

시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

[코드를 사용하지 않고 계속\(W\) →](#)

# 새 프로젝트 만들기

## 최근 프로젝트 템플릿(R)

빈 프로젝트

C++

템플릿 검색(Alt+S)(S)

모든 언어(L)

모든 플랫폼(P)

모든 프로젝트 형식(T)



메이크파일 프로젝트

사용자 빌드 시스템을 가져와 C++를 컴파일하세요.

C++

Windows

데스크톱

콘솔

라이브러리



기본 단위 테스트 프로젝트

기본 Microsoft CppUnitTest 프레임워크를 사용하여 C++ 단위 테스트를 작성합니다.

C++

Windows

테스트



Google Test

Google Test를 사용하여 C++ 단위 테스트를 작성하세요. 사용할 Google Test 라이브러리의 복사본을 포함합니다.

C++

Windows

테스트



내보내기가 있는 동적 연결 라이브러리(DLL)

실행 중인 여러 Windows 앱 간에 공유할 수 있는 .dll을 빌드하세요.

원하는 항목을 찾지 못하셨나요?

[추가 도구 및 기능 설치](#)

뒤로(B)

다음(N)

수정 — Visual Studio Community 2019 — 16.11.27

워크로드   개별 구성 요소   언어 팩   설치 위치



#### Python 개발

Python에 대한 편집, 디버깅, 대화형 개발 및 소스 제어입니다.



#### Node.js 개발

비동기 이벤트 구동 JavaScript 런타임인 Node.js를 사용하여 확장 가능한 네트워크 애플리케이션을 빌드합니다.

### 데스크톱 및 모바일 (5)



#### .NET 데스크톱 개발

.NET 및 .NET Framework와 함께 C#, Visual Basic 및 F#를 사용하여 WPF, Windows Forms 및 콘솔 애플리케이션을...



#### C++를 사용한 데스크톱 개발

MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.



#### 유니버설 Windows 플랫폼 개발

C#, VB 또는 C++(선택 사항)를 사용하여 유니버설 Windows 플랫폼용 애플리케이션을 만듭니다.



#### Mobile development with .NET (out of support)

Xamarin을 사용하여 iOS, Android 또는 Windows용 플랫폼 간 애플리케이션을 빌드합니다.



#### C++를 사용한 모바일 개발

C++를 사용하여 iOS, Android 또는 Windows용 플랫폼 간 애플리케이션을 빌드합니다.

### 설치 세부 정보

#### ▼ C++를 사용한 데스크톱 개발

##### ▼ 포함됨

✓ C++ 핵심 데스크톱 기능

##### ▼ 선택 사항

- ☒ MSVC v142 - VS 2019 C++ x64/x86 빌드...
- ☒ Windows 10 SDK(10.0.19041.0)
- ☒ Just-In-Time 디버거
- ☒ C++ 프로파일링 도구
- ☒ Windows용 C++ CMake 도구
- ☒ 최신 v142 빌드 도구용 C++ ATL(x86 및 x64)
- ☒ Test Adapter for Boost.Test
- ☒ Test Adapter for Google Test
- ☒ Live Share
- ☒ IntelliCode
- ☒ C++ AddressSanitizer
- ☐ MSVC v142 - VS 2019 C++ ARM64 빌드 도...
- ☐ 최신 v142 빌드 도구용 C++ MFC(x86 및 x6...
- ☐ v142 빌드 도구용 C++/CLI 지원(최신)
- ☐ v142 빌드 도구용 C++ 모듈(x64/x86 - 실...

#### 위치

D:\edu\VisualStudio

계속하면 선택한 Visual Studio 버전에 대한 [라이선스](#)에 동의하게 됩니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 [타사 고지 사항](#) 또는 해당 라이선스에 명시된 것처럼 별도로 라이선스가 부여됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

#### 지원되지 않는 구성 요소 제거(R)

시스템 드라이브(C) 0B

다른 드라이브 0B

필요한 전체 공간 0B

다운로드하는 동안 설치

닫기(C)

# 새 프로젝트 만들기

## 최근 프로젝트 템플릿(R)

 빈 프로젝트

C++

템플릿 검색(Alt+S)(S)



모든 언어(L)

모든 플랫폼(P)

모든 프로젝트 형식(T)



빈 프로젝트

Windows용 C++를 사용하여 처음부터 시작합니다. 시작 파일을 제공하지 않습니다.

C++

Windows

콘솔



콘솔 앱

Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.

C++

Windows

콘솔



CMake 프로젝트

.sln 또는 .vcxproj 파일에 종속되지 않은 최신 플랫폼 간 C++ 앱을 빌드하세요.

C++

Windows

Linux

콘솔



Windows 데스크톱 마법사

마법사를 사용하여 고유한 Windows 앱을 만드세요.

C++

Windows

데스크톱

콘솔

라이브러리



Windows 데스크톱 애플리케이션

Windows에서 실행되는 그래픽 사용자 인터페이스를 사용하는 애플리케이션용 프로젝트입니다.

C++

Windows

데스크톱



빈 솔루션

뒤로(B)

다음(N)



# 새 프로젝트 구성

빈 프로젝트

C++

Windows

콘솔

프로젝트 이름(I)

연습\_프로젝트

위치(L)

C:\Users\Worigener\Desktop\c\_주말

...

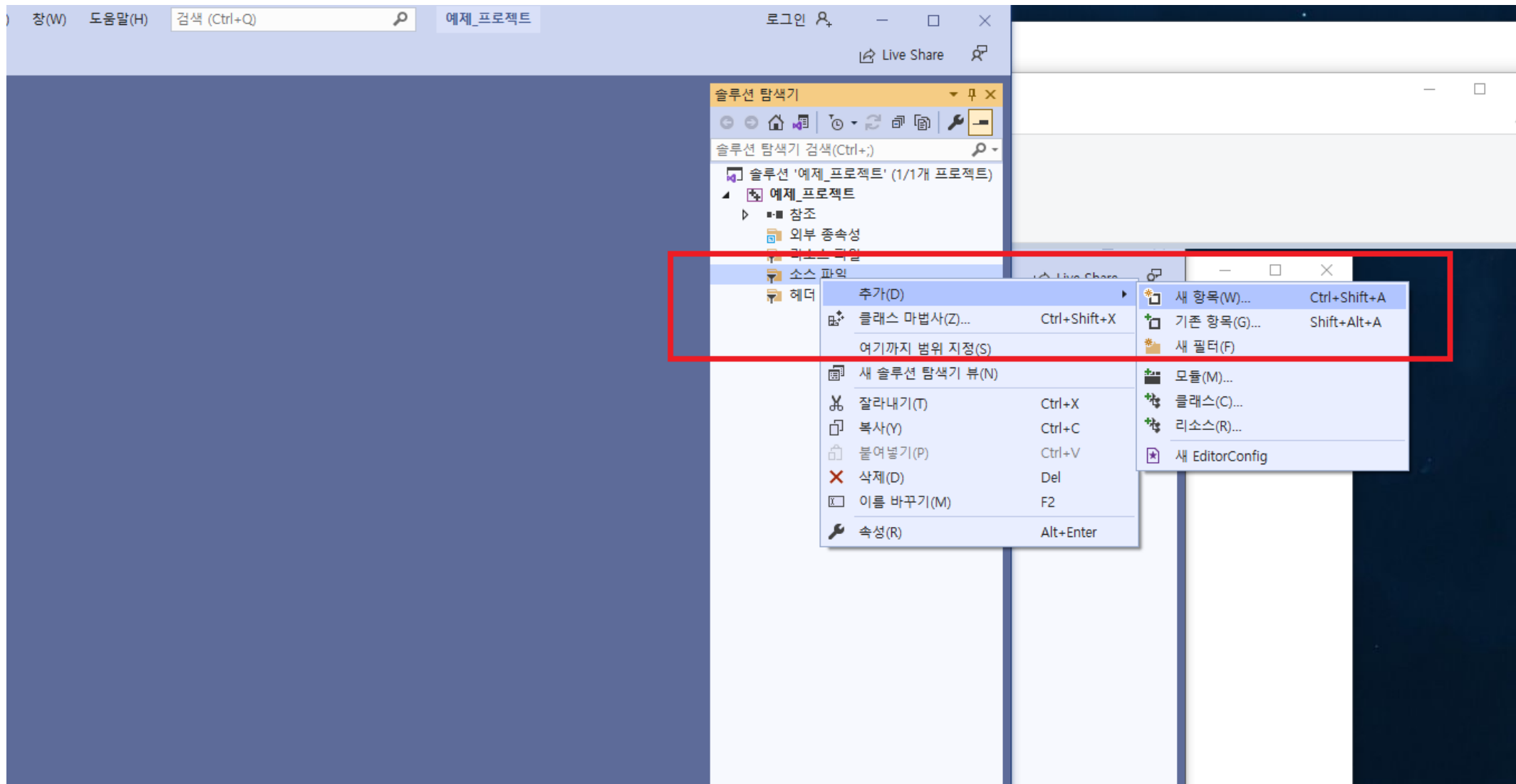
솔루션 이름(M) ⓘ

연습\_프로젝트

☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

뒤로(B)

만들기(C)



새 항목 추가 - 연습\_프로젝트

설치됨

Visual C++

코드

서식

ATL

데이터

리소스

웹

유틸리티

속성 시트

테스트

HLSL

온라인

정렬 기준: 기본값



C++ 파일(.cpp)

Visual C++



헤더 파일(.h)

Visual C++



C++ 클래스

Visual C++



C++ 모듈 인터페이스 단위(.ixx)

Visual C++

검색(Ctrl+E)



유형: Visual C++

C++ 소스 코드를 포함하는 파일을 만듭니다.

연습\_1.c

이름(N):

연습\_1.c

위치(L):

C:\Users\origener\Desktop\c\_주말\연습\_프로젝트\연습\_프로젝트

찾아보기(B)...

추가(A)

취소

# 전체강의 목표

## 학생부 프로그램

```
C:\Users\origener\Desktop\c_주말\연습_프로젝트\학생부_예제\Debug\학생부_예제.exe
메뉴를 선택하세요
(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)
1
1번째 학생데이터를 추가합니다.
이름 : 김학생
점수 : 80
메뉴를 선택하세요
(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)
3
학생데이터를 출력합니다.
-----
1번째 학생
이름 : 김학생, 점수 : 80
-----
메뉴를 선택하세요
(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)
```

# 배울게 될 개념

변수  
연산자  
조건문  
반복문  
배열

```
void st_print(Student* st_p, int count);  
void st_find(Student* st_p, int count);
```

```
int main(void) {  
    int count = 0;  
    Student* st_p;  
    st_p = (Student*)malloc(sizeof(Student) * max_size);  
    Command c;  
    while (1) {  
        puts("메뉴를 선택하세요 \n(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)");  
        scanf_s("%d", &c);  
        switch (c) {  
            case quit:  
                printf("프로그램을 종료합니다.\n");  
                for (int i = 0; i < count; i++) {  
                    free(st_p[i].name);  
                }  
                free(st_p);  
                return 0;  
            case add:  
                st_add(&st_p, &count);  
                break;  
            case find:
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
typedef struct {
    char* name;
    int score;
}Student;
```

```
typedef enum {
    quit,
    add,
    find,
    print,
}Command;
```

```
#define STEP_SIZE 3
#define BUFFER_SIZE 100
int max_size = STEP_SIZE;
```

```
void st_add(Student** st_p, int* count);
void st_print(Student* st_p, int count);
void st_find(Student* st_p, int count);
```

# 구조체 함수 포인터 메모리관리

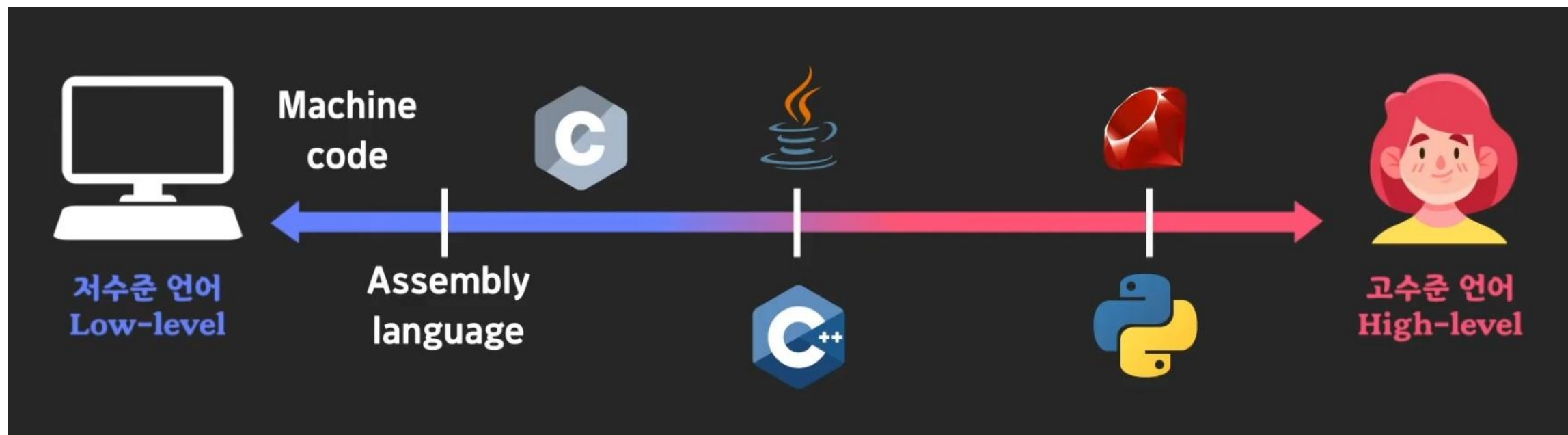
C 언어는 작고 빠르다.

대부분의 환경에서 잘 동작한다.

배우기는 어렵지만 배울 문법이 많이 없다.

사용자 보다는 하드웨어와 가깝다.

개발에 도움을 주는 보조기능들이 별로 없다.





TOUCH SENSITIVE



NETWORK STORAGE



SMARTPHONE OS



FRAMEWORK



CODE TEMPLATE



DOWNLOAD SPEED



DATA TRANSFER



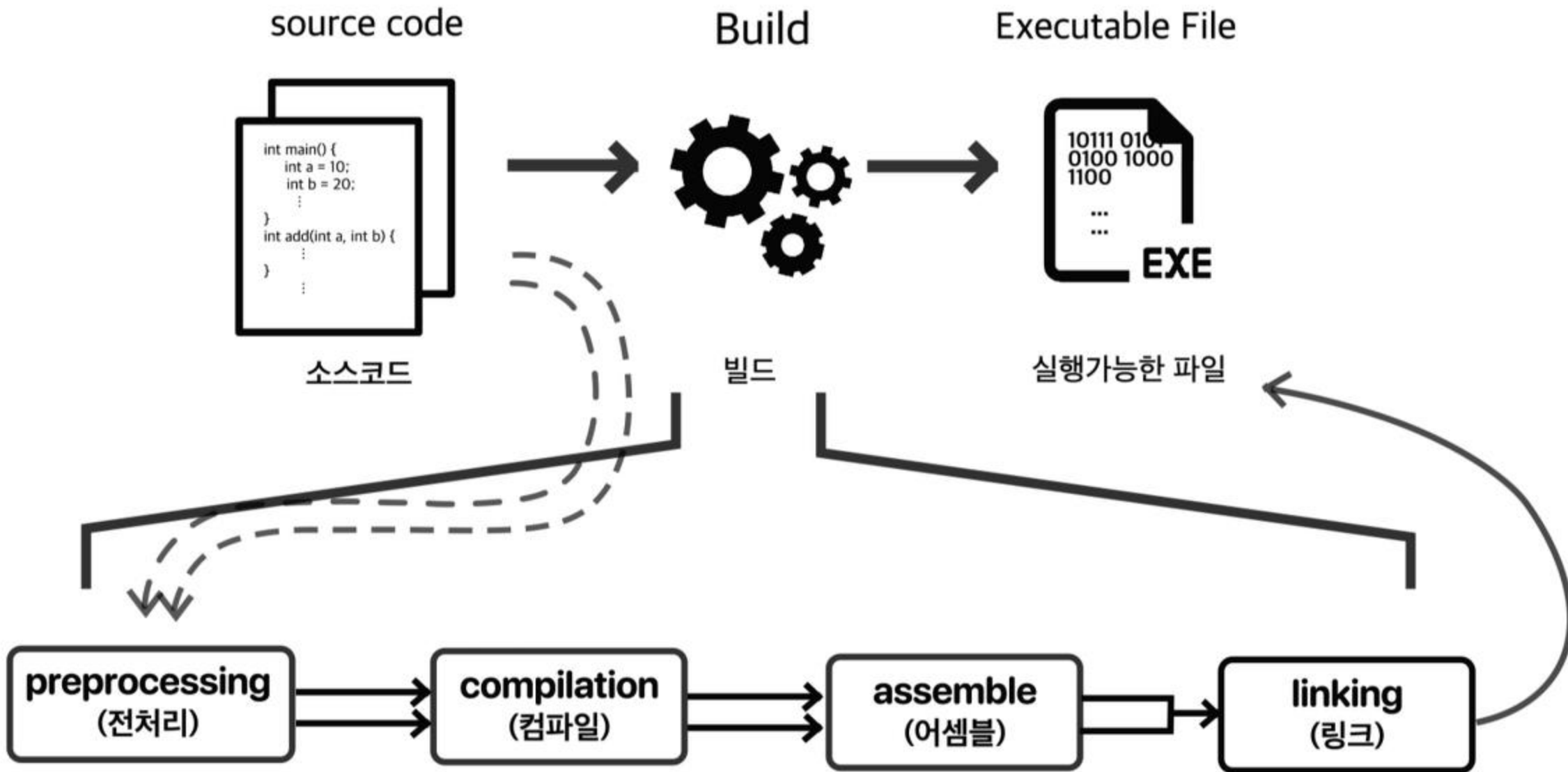
SIGNAL



VIDEO EDITING



# Compile 과정



# 주석 (comments)

프로그램에 영향을 주지않는 정보들

// 한줄짜리 주석

/\*

여러줄의  
주석을 만들때

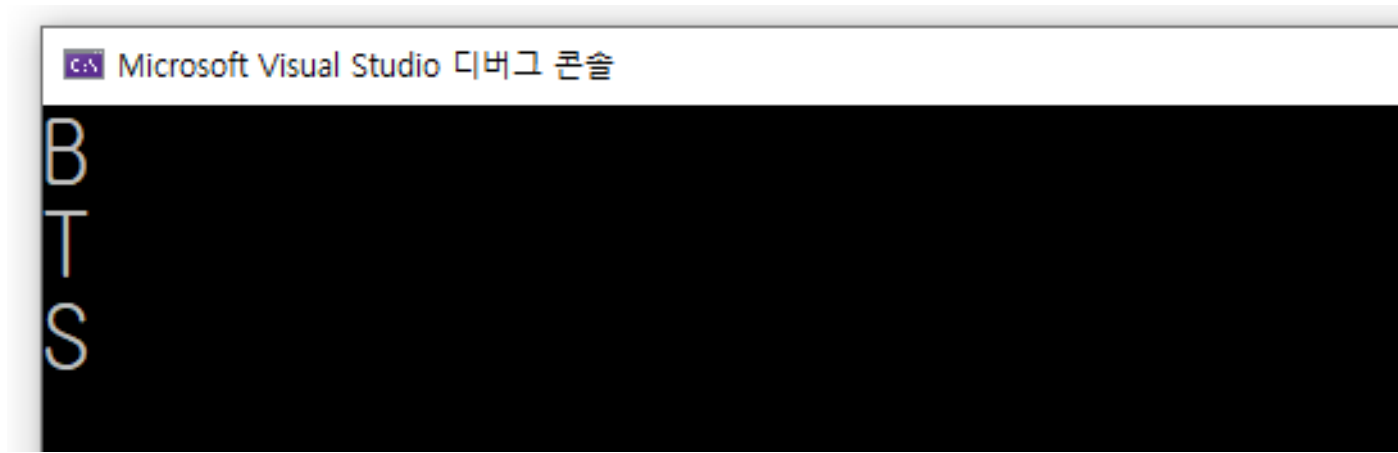
\*/

# Main 함수 (Function)

```
#include <stdio.h>
```

```
int main(void) {  
    printf("hello world! \n");  
    // \n는 한줄을 띄우라는 특수기호  
    return 0;  
}
```

아래와 같이 출력되도록 코드를 작성해보세요.



```
int main(void) {  
    printf("B\nT\nS\n");  
    return 0;  
}
```

```
#include <stdio.h>
```

```
int main(void) {
```

```
    printf("15에서 7을 뺀 값은 %d 입니다.", 15 - 7);
```

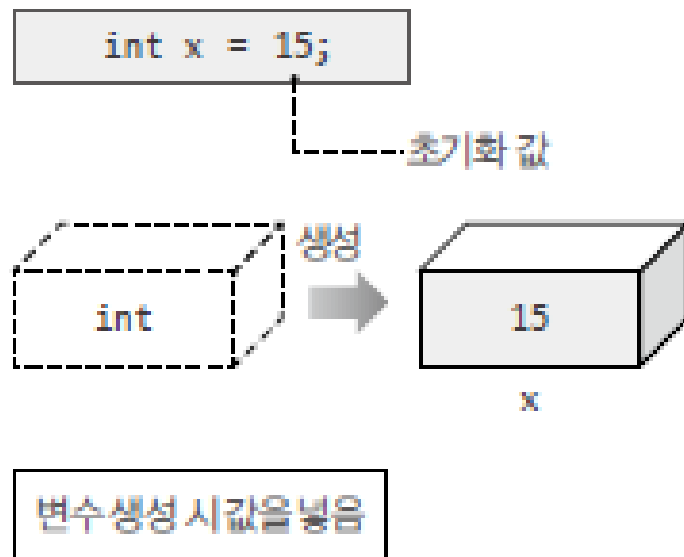
```
    return 0;
```

```
}
```

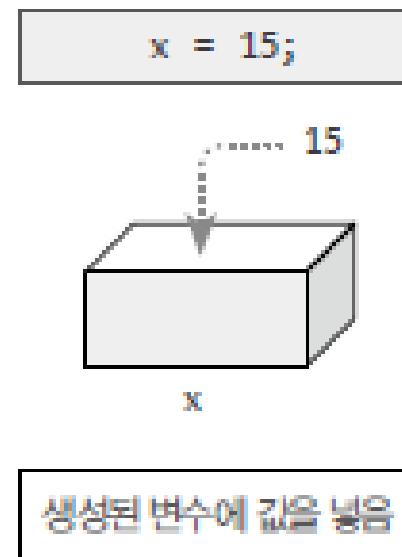
# 변수-p30 (variable)

값이 저장되는  
메모리 주소의  
이름

㉠ 초기화



㉡ 대입



```
#include <stdio.h>
```

```
int main(void) {  
    int x = 15;  
}
```

```
int main(void) {  
    int x;  
    x = 15;  
}
```

# 기본예제 -p33

<https://tttsss77.tistory.com/64>

C 코딩스타일 가이드

```
int main(void) {  
    /* 사용 가능한 변수명 */  
    int num = 3;  
    int number5 = 5;  
    int num_ber = 10;  
    int Number = 15;  
  
    /* 사용 불가능 (컴파일 에러) */  
    int hello boy = 3; // 공백 포함  
    int hi#$ = 5; // 특수 기호  
    int 8number = 8; // 숫자로 시작  
    int short = 3; // 특수기능 키워드  
}
```



## 기본예제 -p33

```
#include <stdio.h>
```

```
int main(void) {  
    int number1 = 3;  
    printf( "%d\n", number1);  
  
    int number2; // 먼저 선언만 하고  
    number2 = 5; // 나중에 값을 할당  
    printf( "%d\n", number2);  
  
    return 0;  
}
```

```
int main(void) {  
    int a = 15, b = 7;
```

아래와 같이 출력되도록 변수 a,b를 이용해서 코드를 작성해보세요.

 Microsoft Visual Studio 디버그 콘솔

a - b 는 8 입니다.

```
int main(void) {  
    int a = 15, b = 7;  
    printf("a - b 는 %d 입니다.", a - b);  
    return 0;  
}
```

```
int main(void) {  
    int a;  
    int b;  
}
```

C:\ 선택 Microsoft Visual Studio 디버그 콘솔

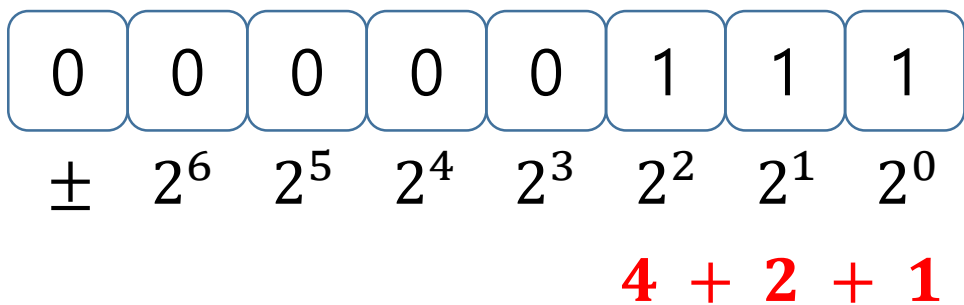
```
a = 10  
b = 4  
a + b는 14 입니다.
```

변수 a,b에 값을 넣고 위그림 처럼 변수를 출력 해보세요.

```
int main(void) {  
    int a;  
    int b;  
  
    a = 10;  
    b = 4;  
    printf("a = %d\n", a);  
    printf("b = %d\n", b);  
    printf("a + b는 %d 입니다.\n", a + b);  
    return 0;  
}
```

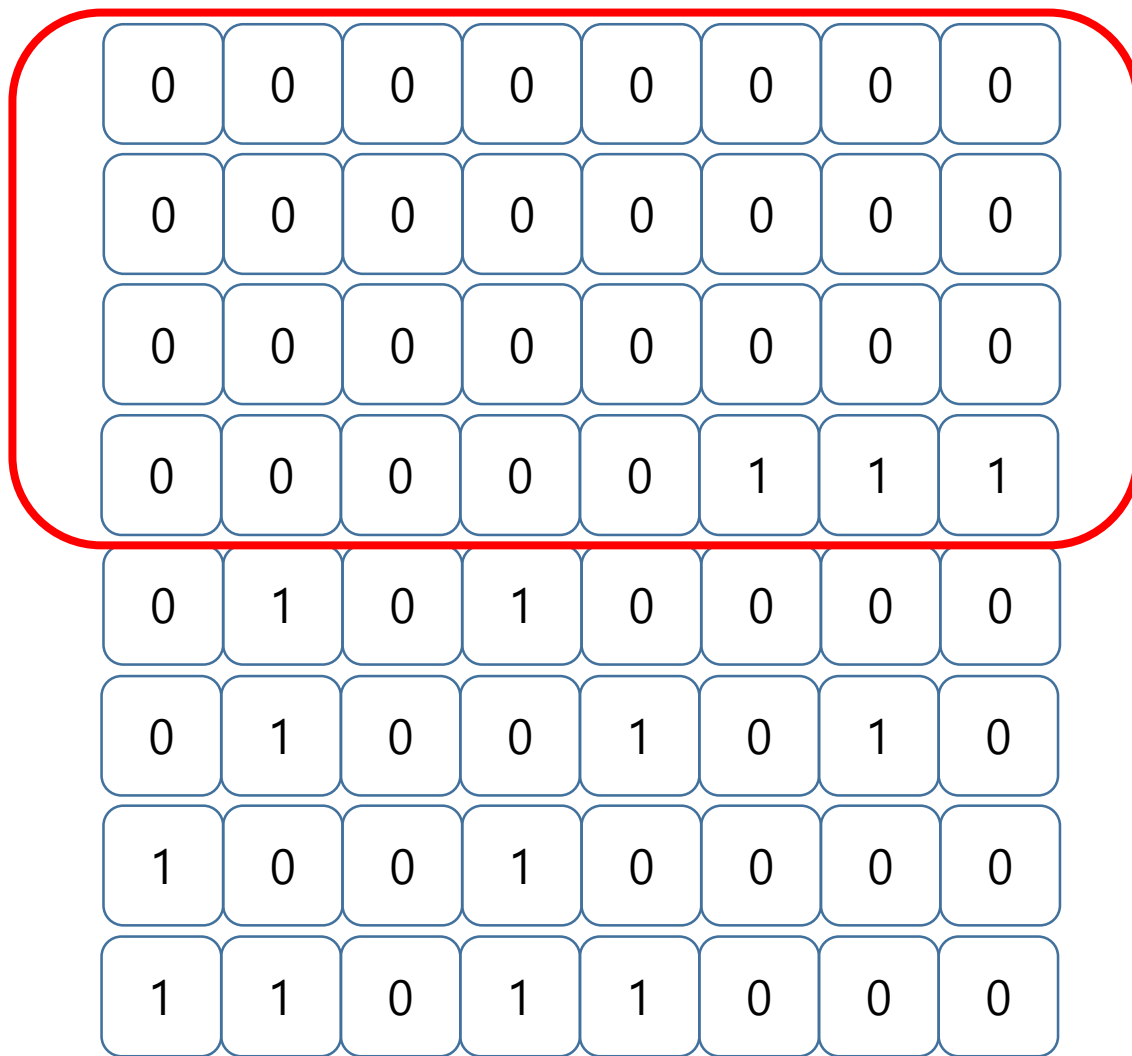
# 변수와 메모리 bit

```
int number = 7;
```



메모리에 저장된 값 : 7

메모리 공간



# 변수와 메모리 주소

```
int main(void) {  
    int number = 7;  
  
    printf("%p", &number);  
    return 0;  
}
```

Microsoft Visual Studio 디버그 콘솔

012FFB54

012FFB54

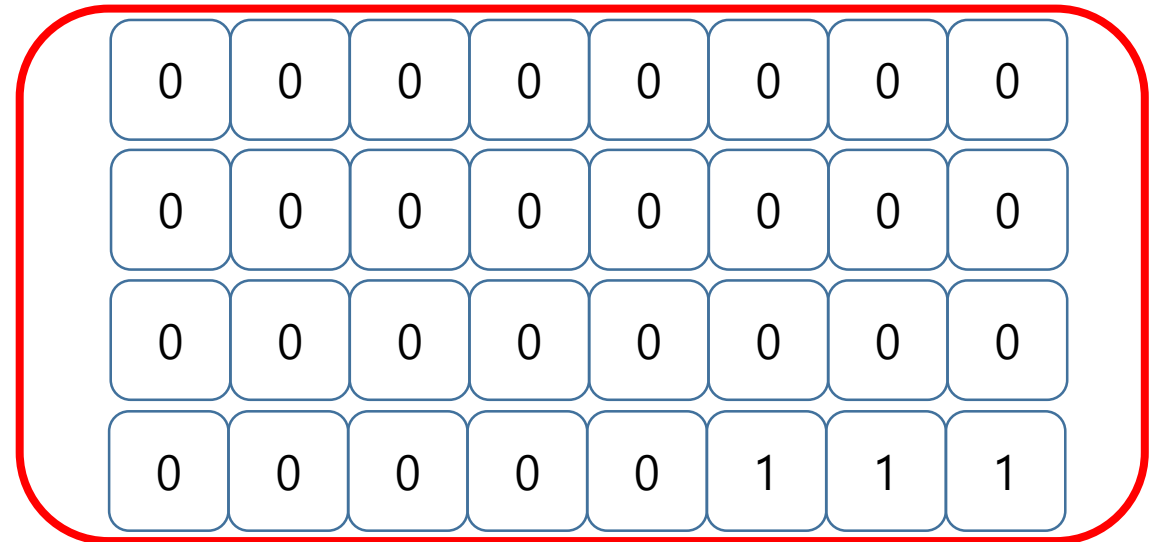
변수 **number**

자료형 : INT

이름 : number

주소 : 012FFB54

값 : 7



# 기본자료형

## P-35

자료형			크기 (byte)	범위
정수	signed	short	2	-32,768 ~ 32,767
		int	4	-2,147,483,648 ~ 2,147,483,647
		long	4	-2,147,483,648 ~ 2,147,483,647
		long long	8	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
	unsigned	unsigned short	2	0 ~ 65,535
		unsigned int	4	0 ~ 4,294,967,295
		unsigned long	4	0 ~ 4,294,967,295
		unsigned long long	8	0 ~ 18,446,744,073,709,551,615
문자	signed	char	1	-128 ~ 127
	unsigned	unsigned char	1	0 ~ 255

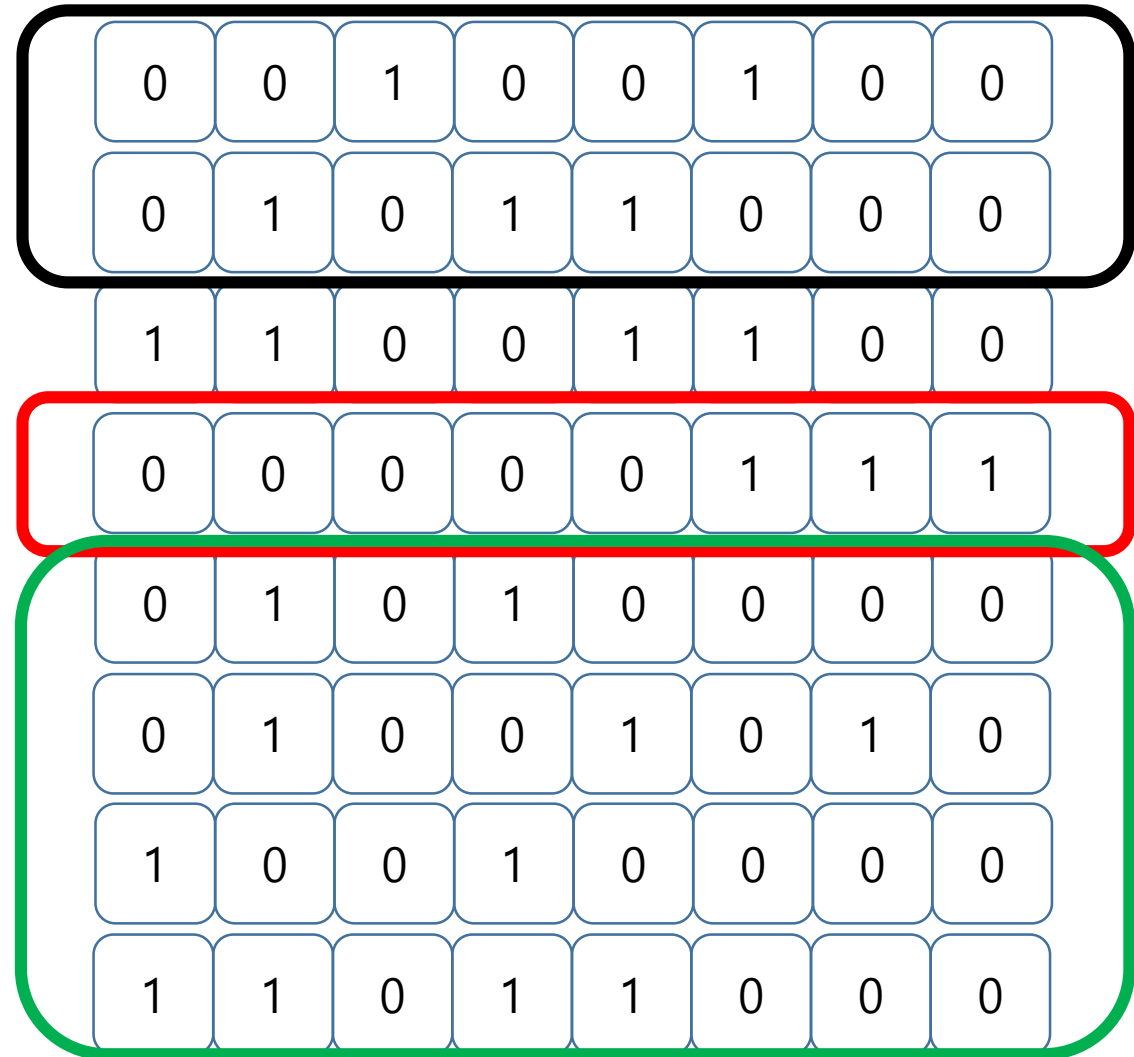
자료형		크기 (byte)	범위	유효 자리수
실수	float	4	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$	소수점 이하 6자리
	double	8	$-1.8 \times 10^{308} \sim 1.8 \times 10^{308}$	소수점 이하 15자리
	long double	8	$-1.8 \times 10^{308} \sim 1.8 \times 10^{308}$	소수점 이하 15자리



```
#include <stdio.h>
```

```
int main(void) {  
    char number = 7;  
    short count;  
    int total;  
    return 0;  
}
```

메모리 공간



```

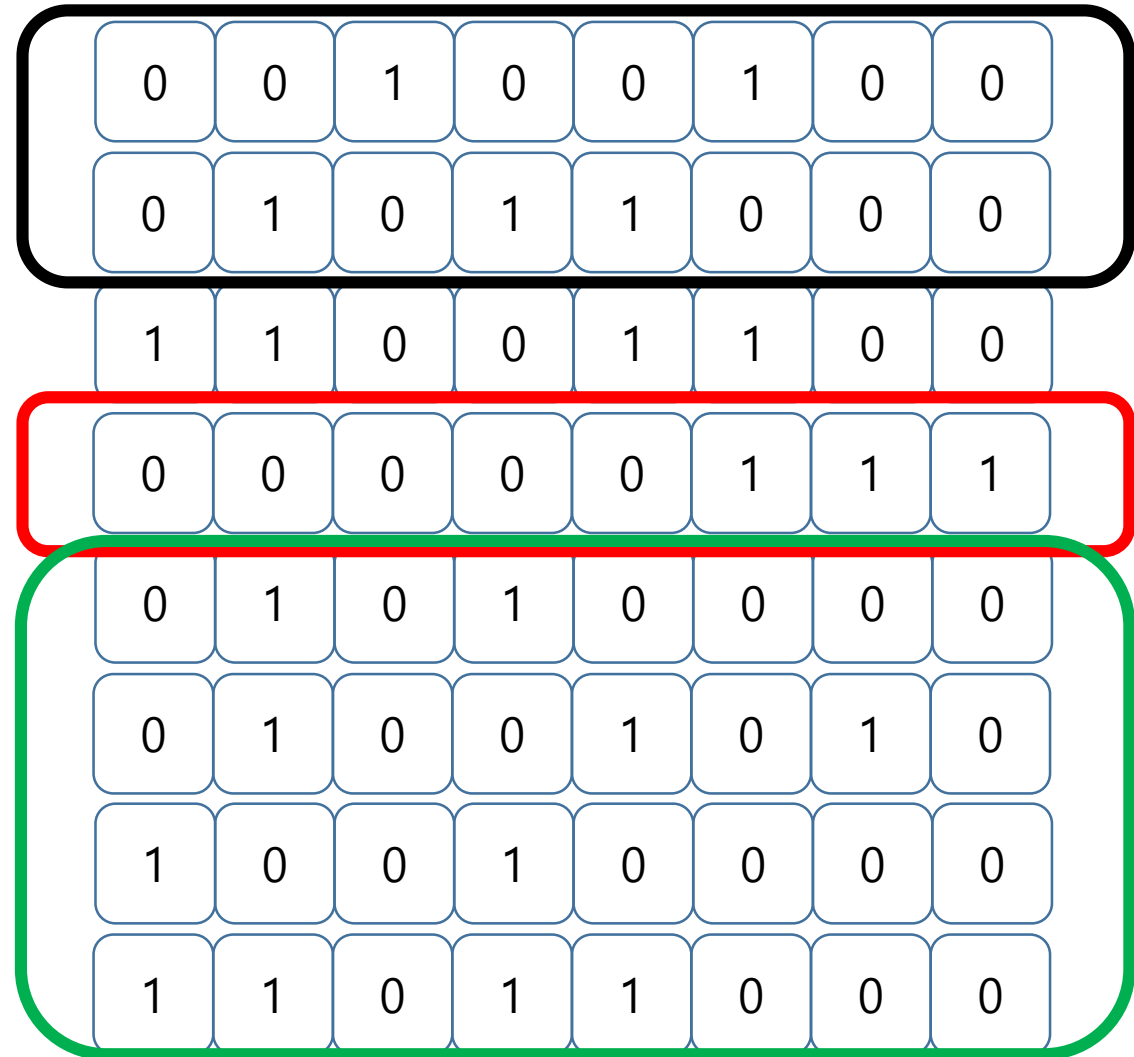
int main(void) {
    char number = 7;
    short count;
    int total;

    printf("%d\n", sizeof(number));
    printf("%d\n", sizeof(count));
    printf("%d\n", sizeof(total));
    return 0;
}

```

# sizeof() 함수 -p36

메모리 공간



# 문자 표현 -p37

041	)	065	A	089	Y	113	q
042	*	066	B	090	Z	114	r
043	+	067	C	091	[	115	s
044	,	068	D	092	\	116	t
045	-	069	E	093	]	117	u
046	.	070	F	094	^	118	v
047	/	071	G	095	_	119	w
048	0	072	H	096	`	120	x
049	1	073	I	097	a	121	y
050	2	074	J	098	b	122	z
051	3	075	K	099	c	123	{
052	4	076	L	100	d	124	
053	5	077	M	101	e	125	}
054	6	078	N	102	f	126	~
055	7	079	O	103	g		

```
int main(void) {  
    char ch1 = 66, ch2 = 'B';  
    short sh1 = 67;  
    int in1 = 68;  
  
    printf( "%c\n", ch1 );  
    printf( "%c\n", ch2 );  
    printf( "%c\n", sh1 );  
    printf( "%c\n", in1 );  
    return 0;  
}
```

ASCII 코드

# 상수 -p41 (constant)

리터럴 (Literal) - 고정값

10, -10, 2.2, 'A', '하늘', 'BTS' ...

```
#include <stdio.h>
```

```
#define LENGTH 10
```

매크로 상수

```
int main(void) {
```

```
    : const int NUMBER = 5;
```

심볼릭 상수

아래와 같이 출력되도록 매크로상수를 이용해서 코드를 작성해보세요.

```
#include <stdio.h>
#define MAX 100
```

```
int main(void) {
```

Microsoft Visual Studio 디버그 콘솔

상수 MAX 값은 100 입니다.

```
}
```

```
#include <stdio.h>
```

```
#define MAX 100
```

```
int main(void) {
```

```
    printf("상수 MAX 값은 %d 입니다.", MAX);
```

```
    return 0;
```

```
}
```

# 기본예제 -p41

```
#include <stdio.h>
#define LENGTH 10
```

```
int main(void) {
    int number = 3;
    const int NUMBER = 5;
    number = 10;
    NUMBER = 10;

    printf("%d\n", LENGTH);
    printf("%d\n", number);
    printf("%d\n", NUMBER);

    return 0;
}
```

# 상수 접미사

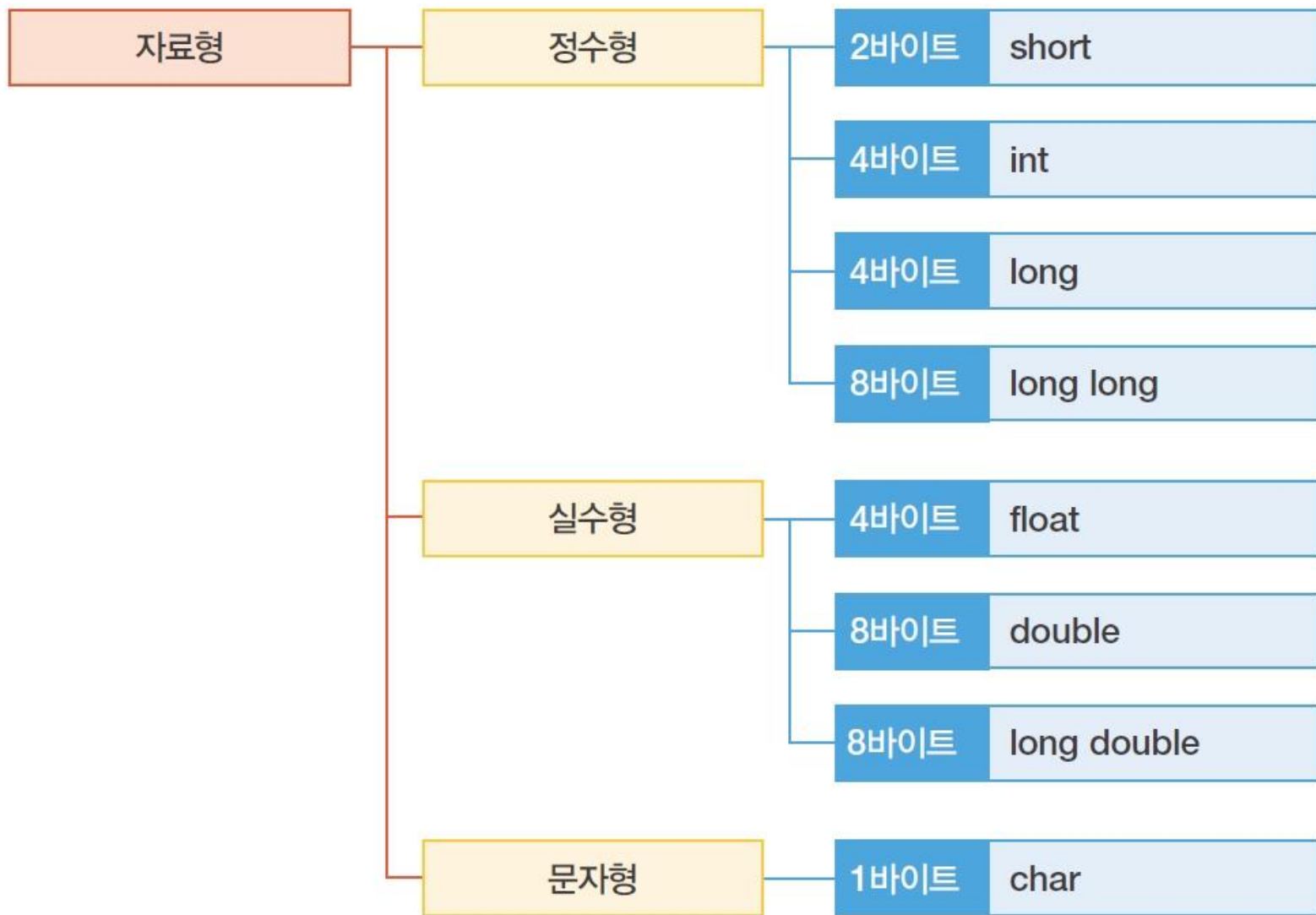
## -p39

리터럴 상수(Literal) - 고정값

10, -10, 2.2, 'A', '하늘', 'BTS' ...

```
int main(void) {  
    long a = 10L;  
    long long b = 100LL;  
  
    float f = 2.2F;  
    long double d = 10.12L;  
  
    printf("데이터 메모리 사이즈 : %d\n", sizeof(10.0F));  
}
```





# <stdint.h>

호환성을 위해

사용되는 비트에 따른

정수형 데이터타입

```
#include <stdio.h>
#include <stdint.h>
```

```
int main(void) {
    int8_t  int_8;
    int16_t int_16;
    int32_t int_32;
    int64_t int_64;

    printf( "%d\n", sizeof( int_8 ) );
    printf( "%d\n", sizeof( int_16 ) );
    printf( "%d\n", sizeof( int_32 ) );
    printf( "%d\n", sizeof( int_64 ) );
}
```

# <stdint.h>

```
#include <stdio.h>
#include <stdint.h>
```

각각의 데이터타입이

표현할 수 있는 최대 / 최소값이

매크로상수로 정의되어있음

```
int main(void) {
    printf("max: %d / min: %d\n", INT8_MAX, INT8_MIN);
    printf("max: %lld / min %lld\n", INT64_MAX, INT64_MIN);
}
```

# 형변환 -p43

```
int main(void) {  
    double num1 = 10;  
    int num2 = 1.2345;  
    short num3 = 70000;  
  
    printf( "%f\n", num1);  
    printf( "%d\n", num2);  
    printf( "%d\n", num3);  
    return 0;  
}
```

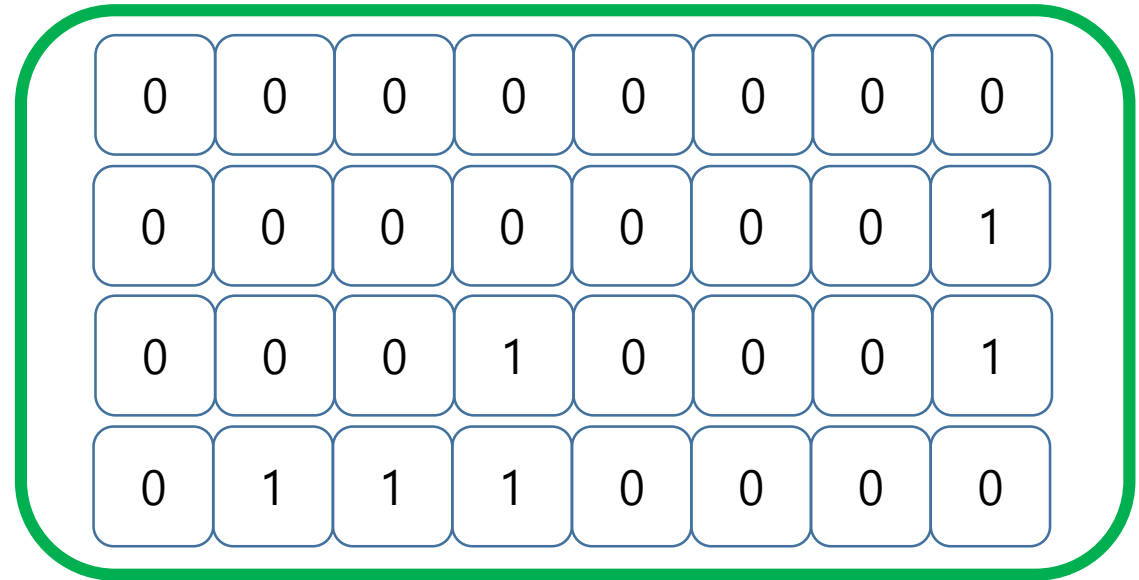
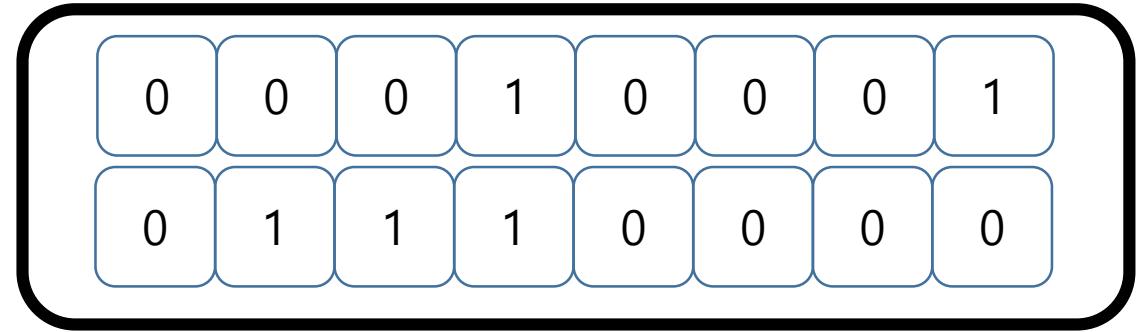
```
int main(void) {
```

```
    short num = 70000;
```

```
    printf("%d\n", num);
```

```
    return 0;
```

```
}
```



# 형변환 -p43

```
int main(void) {  
    printf( "%d\n", (short)3.1415);  
    printf( "%d\n", (int)3.1415);  
    printf( "%f\n", (double)10);  
    printf( "%f\n", (float)10);  
    return 0;  
}
```

# 응용문제 p-45

다음코드는 컴파일 에러를 발생시키지는 않지만, 아래와같은 경고 메시지가 나타납니다.

그 이유를 생각해보고 해결책을 제시해 보세요

```
int main(void) {  
    float f = 3.14;  
  
    printf("%f\n", f);  
}
```

warning C4305: '초기화 중': 'double'에서 'float'(으)로 잘립니다.

# •진수표현 – p48

다른진수를 10진법 으로 변환하는 방법 (손으로)

10진수	2진수	8진수	16진수
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
20	10100	24	14



다른진수를 10진법 으로 변환하는 방법 (c문법)

```
int main(void) {  
    int binary_10 = 0b1010;  
    int octal_10 = 012;  
    int hexa_decimal_10 = 0xA;  
  
    printf("2진수->10진수 : %d\n", binary_10);  
    printf("8진수->10진수 : %d\n", octal_10);  
    printf("16진수->10진수 : %d\n", hexa_decimal_10);  
}
```

10진수	2진수	8진수	16진수
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A

10진수를 다른 진법으로 변환하는 방법 (손으로)

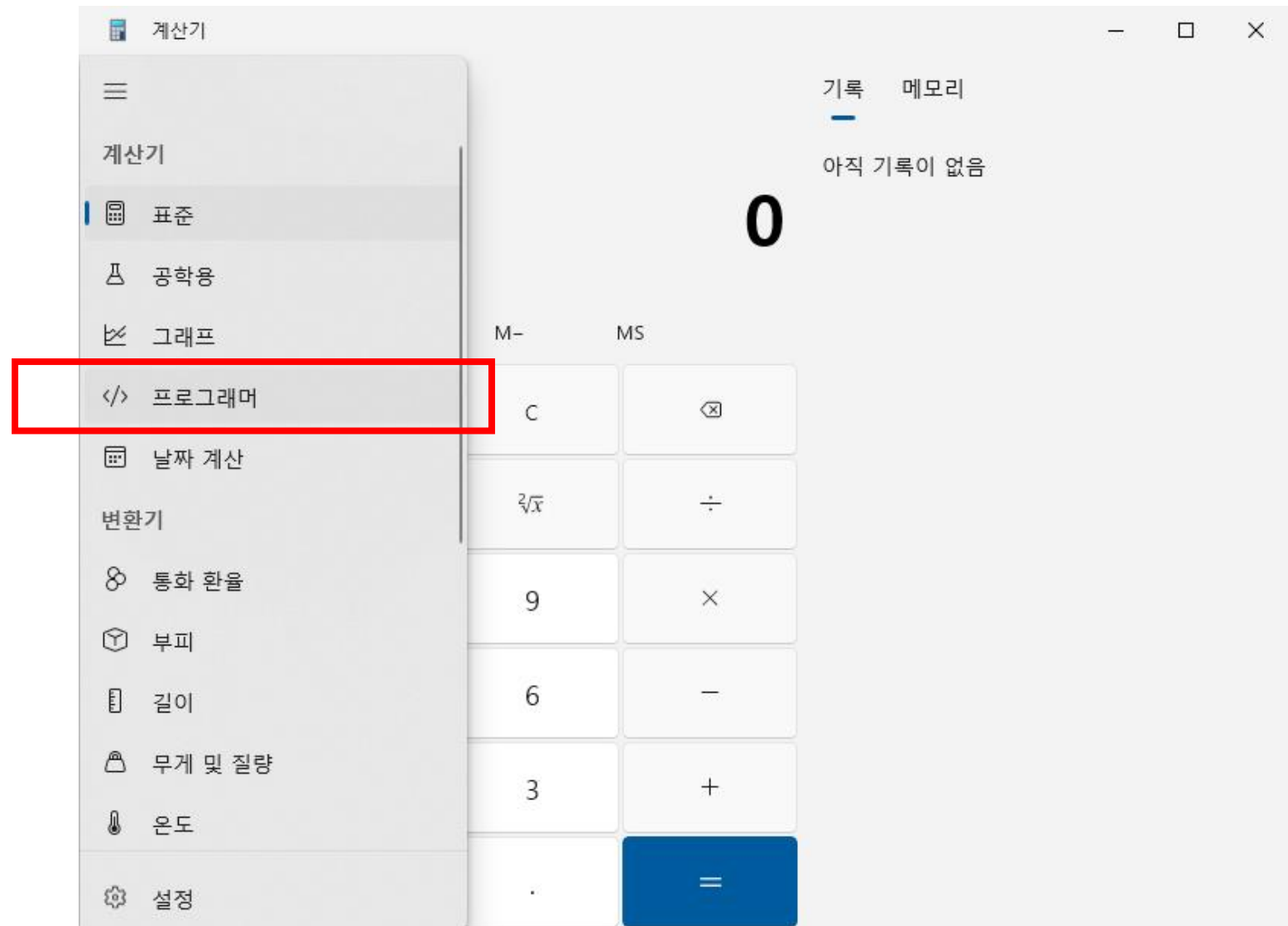
2	47	나머지
2	23	1
2	11	1
2	5	1
2	2	1
	1	0



10진수를 다른 진법으로 변환하는 방법 (c문법)

```
int main(void) {  
    printf("777 -> 8진수 : %o\n", 777);  
    printf("777 -> 16진수 : %x\n", 777);  
}
```

## 계산기로 진수 변환 하는법



## 정수의 음수표현 - 2의 보수

```
int main(void) {  
    char number = 130;  
    unsigned char u_number = 130;  
    printf( "%d\n", number );  
    printf( "%d\n", u_number );  
    return 0;  
}
```

1	0	0	0	0	0	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

$$= -126$$

## 정수의 음수표현 - 2의 보수

1	0	0	0	0	0	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

= -2 아닌가???

+

0	0	0	0	0	0	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

= 2

||

1	0	0	0	0	1	0	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

$\neq 0$

## 정수의 음수표현 - 2의 보수

1	0	0	0	0	0	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

= -2 아님 -126임

+

0	1	1	1	1	1	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

= 126

||

0	0	0	0	0	0	0	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

= 0

## 정수의 음수표현 - 2의 보수

1. 마이너스 부호를 때고 이진수로

0	1	1	1	1	1	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

-126  $\rightarrow$  126

↓ 2. 모든 bit를 반전시킨다

1	0	0	0	0	0	0	1
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

↓ 3. +1을 더한다

1	0	0	0	0	0	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

= -126



## 음수 비트를 십진수로

1	0	1	1	0	0	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

↓ 1. 비트에서 1을 뺀다

1	0	1	1	0	0	0	1
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

↓ 2. 모든 비트를 반전시킨다

0	1	0	0	1	1	1	0
$\pm$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

= 78 → -78

3. 마이너스 부호 추가

## 응용 문제 -p56

다음 비트들을 10진수로 표현해 보세요

1.

01101001

2.

01011001

3.

00110100

## 응용 문제 -p56

다음 비트들을 10진수로 표현해 보세요

1.

$$\begin{array}{ccccccc} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ \pm & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} = 64 + 32 + 8 + 1 = 105$$

2.

$$\begin{array}{ccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ \pm & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} = 64 + 16 + 8 + 1 = 89$$

3.

$$\begin{array}{ccccccc} 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ \pm & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} = 32 + 16 + 4 = 56$$

# 응용 문제

다음 비트들을 16진수로 표현해 보세요

1.

01111011

2.

01000001

3.

00110111

## 응용 문제 -p56

다음 음수 비트들을 10진수로 표현해 보세요

1.

10101001

2.

11110000

3.

10001010

## 응용 문제 -p56

다음 음수 비트들을 10진수로 표현해 보세요

1.

$$10101001 \xrightarrow{(-1)} 10101000 \xrightarrow{\text{반전}} 01010111 = -87$$

2.

$$11110000 \xrightarrow{(-1)} 11101111 \xrightarrow{\text{반전}} 00010000 = -16$$

3.

$$10001010 \xrightarrow{(-1)} 10001001 \xrightarrow{\text{반전}} 01110110 = -118$$